

### **In the Specification**

Please amend the paragraph beginning on page 12, line 8 as follows:

If the words to be sorted are strings characterized by a variable string length, then the execution time is proportional to  $\sum_j W_j N_j$ , where  $N_j$  is a string length in bits or bytes (assuming that the number of bits per byte is a constant), wherein  $W_j$  is a weighting factor that is proportional to the number of strings to be sorted having a string length  $N_j$ . The summation  $\sum_j$  is from  $j=1$  to  $j=J$  such that  $J$  is the number of unique string lengths in the strings to be sorted. For example consider 60 strings to be sorted such that 30 strings have 3 bytes each, 18 strings have 4 bytes each, and 12 strings have 5 bytes each. For this example,  $J=3$ ,  $N_1=3$  bytes,  $W_1 \propto 30$ ,  $N_2=4$  bytes,  $W_2 \propto 18$ ,  $N_3=5$  bytes,  $W_3 \propto 12$  bytes, wherein the symbol " $\propto$ " stands for "proportional to". Thus, the sort execution time is a linear combination of the string lengths  $N_j$  (expressed in bits or bytes) of the variable-length strings to be sorted. Accordingly, the sort algorithm of the present invention is properly designated herein as a "linear sort" for the case of sorting variable-length strings.

Please amend the paragraph beginning on page 40, line 21 as follows:

Also note that although FIG. 6 expresses program logic [[to]] natural to counter-controlled looping through the program code, while FIG. 5 expresses logic natural to recursive execution of the program code, the fundamental method of sorting of the present invention and the associated key steps thereof are essentially the same in FIGS. 5 and 6. Thus, the logic depicted in FIG. 6 is merely illustrative, and the counter-controlled looping embodiment may be implemented in any manner that would be apparent to an ordinary person in the art of computer programming who is

familiar with the fundamental sorting algorithm described herein. As an example, the counter-controlled looping embodiment may be implemented in a manner that parallels the logic of FIG. 5 with the exceptions of: 1) the counter-controlled looping through the program code replaces the recursive execution of the program code; and 2) counters associated with the counter-controlled looping need to be programmatically tracked, updated, and tested.